

1/13

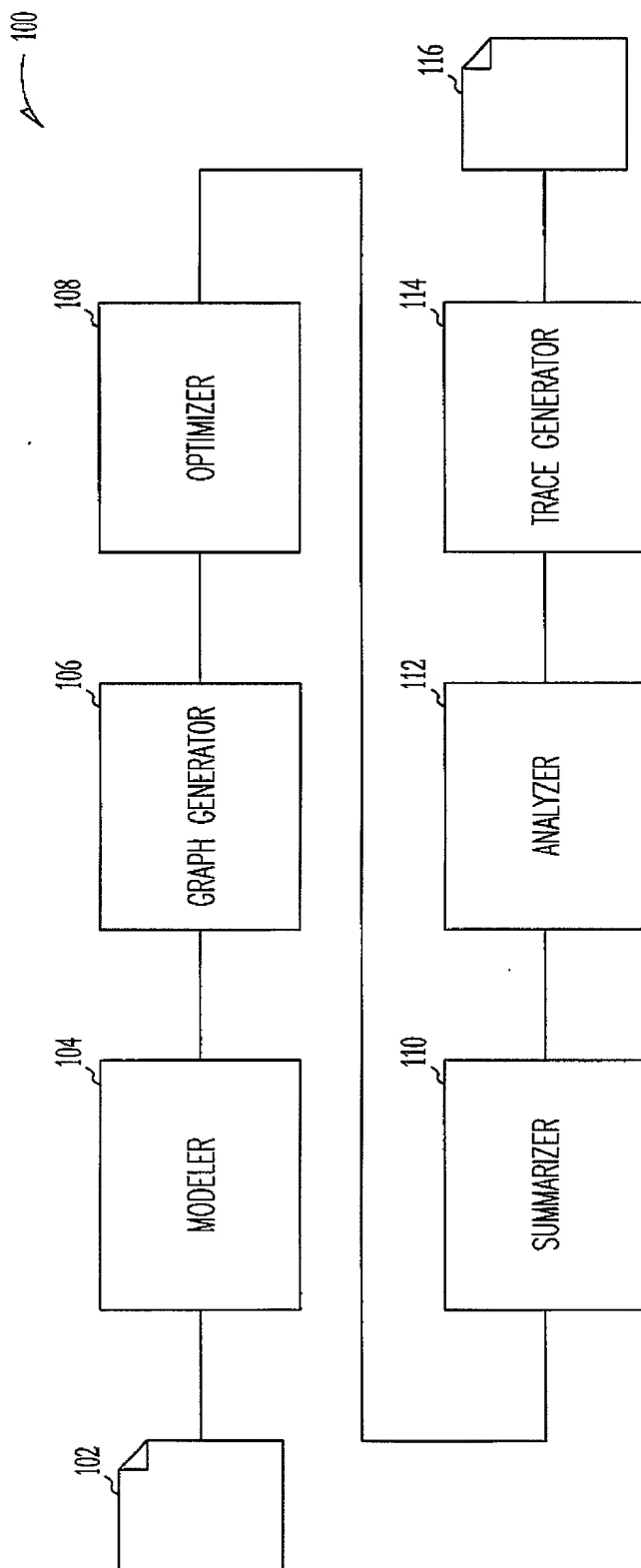


Fig.1

00 — A

```

int numUnits;
int level;
void getUnit() {
    [1] bool canEnter := F;
    [2] if (numUnits = 0) {
    [3]   if (level > 10) {
    [4]     NewUnit();
    [5]     numUnits := 1;
    [6]     canEnter := T;
    [7]   } else
    [8]     canEnter := T;
    [9]   if (canEnter)
    [10]    if (numUnits = 0) {
    [11]      assert(F);
    [12]    } else
    [13]      gotUnit();
    [14] }
    [15] }

    [16] ...;
    [17] if (?) {
    [18]   if (?) {
    [19]     ...;
    [20]     numUnits := F;
    [21]     ...;
    [22]   } else
    [23]     ...;
    [24] }
    [25] ...;
    [26] if (nU0) {
    [27]   if (?) {
    [28]     ...;
    [29]     nU0 := F;
    [30]     ...;
    [31]   } else
    [32]     ...;
    [33] }
    [34] ...;
    [35] if (cE) {
    [36]   if (nU0) {
    [37]     ...;
    [38]     nU0 := F;
    [39]     cE := T;
    [40]   } else
    [41]     cE := T;
    [42] }
    [43] ...;
    [44] if (cE)
    [45]   if (nU0)
    [46]     ...;
    [47]   else
    [48]     ...;
    [49] }
    [50] }
    [51] }
    [52] }
    [53] }
    [54] }
    [55] }
    [56] }
    [57] }
    [58] }
    [59] }
    [60] }
    [61] }
    [62] }
    [63] }
    [64] }
    [65] }
    [66] }
    [67] }
    [68] }
    [69] }
    [70] }
    [71] }
    [72] }
    [73] }
    [74] }
    [75] }
    [76] }
    [77] }
    [78] }
    [79] }
    [80] }
    [81] }
    [82] }
    [83] }
    [84] }
    [85] }
    [86] }
    [87] }
    [88] }
    [89] }
    [90] }
    [91] }
    [92] }
    [93] }
    [94] }
    [95] }
    [96] }
    [97] }
    [98] }
    [99] }
    [100] }
    [101] }
    [102] }
    [103] }
    [104] }
    [105] }
    [106] }
    [107] }
    [108] }
    [109] }
    [110] }
    [111] }
    [112] }
    [113] }
    [114] }
    [115] }
    [116] }
    [117] }
    [118] }
    [119] }
    [120] }
    [121] }
    [122] }
    [123] }
    [124] }
    [125] }
    [126] }
    [127] }
    [128] }
    [129] }
    [130] }
    [131] }
    [132] }
    [133] }
    [134] }
    [135] }
    [136] }
    [137] }
    [138] }
    [139] }
    [140] }
    [141] }
    [142] }
    [143] }
    [144] }
    [145] }
    [146] }
    [147] }
    [148] }
    [149] }
    [150] }
    [151] }
    [152] }
    [153] }
    [154] }
    [155] }
    [156] }
    [157] }
    [158] }
    [159] }
    [160] }
    [161] }
    [162] }
    [163] }
    [164] }
    [165] }
    [166] }
    [167] }
    [168] }
    [169] }
    [170] }
    [171] }
    [172] }
    [173] }
    [174] }
    [175] }
    [176] }
    [177] }
    [178] }
    [179] }
    [180] }
    [181] }
    [182] }
    [183] }
    [184] }
    [185] }
    [186] }
    [187] }
    [188] }
    [189] }
    [190] }
    [191] }
    [192] }
    [193] }
    [194] }
    [195] }
    [196] }
    [197] }
    [198] }
    [199] }
    [200] }
    [201] }
    [202] }
    [203] }
    [204] }
    [205] }
    [206] }
    [207] }
    [208] }
    [209] }
    [210] }
    [211] }
    [212] }
    [213] }
    [214] }
    [215] }
    [216] }
    [217] }
    [218] }
    [219] }
    [220] }
    [221] }
    [222] }
    [223] }
    [224] }
    [225] }
    [226] }
    [227] }
    [228] }
    [229] }
    [230] }
    [231] }
    [232] }
    [233] }
    [234] }
    [235] }
    [236] }
    [237] }
    [238] }
    [239] }
    [240] }
    [241] }
    [242] }
    [243] }
    [244] }
    [245] }
    [246] }
    [247] }
    [248] }
    [249] }
    [250] }
    [251] }
    [252] }
    [253] }
    [254] }
    [255] }
    [256] }
    [257] }
    [258] }
    [259] }
    [260] }
    [261] }
    [262] }
    [263] }
    [264] }
    [265] }
    [266] }
    [267] }
    [268] }
    [269] }
    [270] }
    [271] }
    [272] }
    [273] }
    [274] }
    [275] }
    [276] }
    [277] }
    [278] }
    [279] }
    [280] }
    [281] }
    [282] }
    [283] }
    [284] }
    [285] }
    [286] }
    [287] }
    [288] }
    [289] }
    [290] }
    [291] }
    [292] }
    [293] }
    [294] }
    [295] }
    [296] }
    [297] }
    [298] }
    [299] }
    [300] }
    [301] }
    [302] }
    [303] }
    [304] }
    [305] }
    [306] }
    [307] }
    [308] }
    [309] }
    [310] }
    [311] }
    [312] }
    [313] }
    [314] }
    [315] }
    [316] }
    [317] }
    [318] }
    [319] }
    [320] }
    [321] }
    [322] }
    [323] }
    [324] }
    [325] }
    [326] }
    [327] }
    [328] }
    [329] }
    [330] }
    [331] }
    [332] }
    [333] }
    [334] }
    [335] }
    [336] }
    [337] }
    [338] }
    [339] }
    [340] }
    [341] }
    [342] }
    [343] }
    [344] }
    [345] }
    [346] }
    [347] }
    [348] }
    [349] }
    [350] }
    [351] }
    [352] }
    [353] }
    [354] }
    [355] }
    [356] }
    [357] }
    [358] }
    [359] }
    [360] }
    [361] }
    [362] }
    [363] }
    [364] }
    [365] }
    [366] }
    [367] }
    [368] }
    [369] }
    [370] }
    [371] }
    [372] }
    [373] }
    [374] }
    [375] }
    [376] }
    [377] }
    [378] }
    [379] }
    [380] }
    [381] }
    [382] }
    [383] }
    [384] }
    [385] }
    [386] }
    [387] }
    [388] }
    [389] }
    [390] }
    [391] }
    [392] }
    [393] }
    [394] }
    [395] }
    [396] }
    [397] }
    [398] }
    [399] }
    [400] }
    [401] }
    [402] }
    [403] }
    [404] }
    [405] }
    [406] }
    [407] }
    [408] }
    [409] }
    [410] }
    [411] }
    [412] }
    [413] }
    [414] }
    [415] }
    [416] }
    [417] }
    [418] }
    [419] }
    [420] }
    [421] }
    [422] }
    [423] }
    [424] }
    [425] }
    [426] }
    [427] }
    [428] }
    [429] }
    [430] }
    [431] }
    [432] }
    [433] }
    [434] }
    [435] }
    [436] }
    [437] }
    [438] }
    [439] }
    [440] }
    [441] }
    [442] }
    [443] }
    [444] }
    [445] }
    [446] }
    [447] }
    [448] }
    [449] }
    [450] }
    [451] }
    [452] }
    [453] }
    [454] }
    [455] }
    [456] }
    [457] }
    [458] }
    [459] }
    [460] }
    [461] }
    [462] }
    [463] }
    [464] }
    [465] }
    [466] }
    [467] }
    [468] }
    [469] }
    [470] }
    [471] }
    [472] }
    [473] }
    [474] }
    [475] }
    [476] }
    [477] }
    [478] }
    [479] }
    [480] }
    [481] }
    [482] }
    [483] }
    [484] }
    [485] }
    [486] }
    [487] }
    [488] }
    [489] }
    [490] }
    [491] }
    [492] }
    [493] }
    [494] }
    [495] }
    [496] }
    [497] }
    [498] }
    [499] }
    [500] }
    [501] }
    [502] }
    [503] }
    [504] }
    [505] }
    [506] }
    [507] }
    [508] }
    [509] }
    [510] }
    [511] }
    [512] }
    [513] }
    [514] }
    [515] }
    [516] }
    [517] }
    [518] }
    [519] }
    [520] }
    [521] }
    [522] }
    [523] }
    [524] }
    [525] }
    [526] }
    [527] }
    [528] }
    [529] }
    [530] }
    [531] }
    [532] }
    [533] }
    [534] }
    [535] }
    [536] }
    [537] }
    [538] }
    [539] }
    [540] }
    [541] }
    [542] }
    [543] }
    [544] }
    [545] }
    [546] }
    [547] }
    [548] }
    [549] }
    [550] }
    [551] }
    [552] }
    [553] }
    [554] }
    [555] }
    [556] }
    [557] }
    [558] }
    [559] }
    [560] }
    [561] }
    [562] }
    [563] }
    [564] }
    [565] }
    [566] }
    [567] }
    [568] }
    [569] }
    [570] }
    [571] }
    [572] }
    [573] }
    [574] }
    [575] }
    [576] }
    [577] }
    [578] }
    [579] }
    [580] }
    [581] }
    [582] }
    [583] }
    [584] }
    [585] }
    [586] }
    [587] }
    [588] }
    [589] }
    [590] }
    [591] }
    [592] }
    [593] }
    [594] }
    [595] }
    [596] }
    [597] }
    [598] }
    [599] }
    [600] }
    [601] }
    [602] }
    [603] }
    [604] }
    [605] }
    [606] }
    [607] }
    [608] }
    [609] }
    [610] }
    [611] }
    [612] }
    [613] }
    [614] }
    [615] }
    [616] }
    [617] }
    [618] }
    [619] }
    [620] }
    [621] }
    [622] }
    [623] }
    [624] }
    [625] }
    [626] }
    [627] }
    [628] }
    [629] }
    [630] }
    [631] }
    [632] }
    [633] }
    [634] }
    [635] }
    [636] }
    [637] }
    [638] }
    [639] }
    [640] }
    [641] }
    [642] }
    [643] }
    [644] }
    [645] }
    [646] }
    [647] }
    [648] }
    [649] }
    [650] }
    [651] }
    [652] }
    [653] }
    [654] }
    [655] }
    [656] }
    [657] }
    [658] }
    [659] }
    [660] }
    [661] }
    [662] }
    [663] }
    [664] }
    [665] }
    [666] }
    [667] }
    [668] }
    [669] }
    [670] }
    [671] }
    [672] }
    [673] }
    [674] }
    [675] }
    [676] }
    [677] }
    [678] }
    [679] }
    [680] }
    [681] }
    [682] }
    [683] }
    [684] }
    [685] }
    [686] }
    [687] }
    [688] }
    [689] }
    [690] }
    [691] }
    [692] }
    [693] }
    [694] }
    [695] }
    [696] }
    [697] }
    [698] }
    [699] }
    [700] }
    [701] }
    [702] }
    [703] }
    [704] }
    [705] }
    [706] }
    [707] }
    [708] }
    [709] }
    [710] }
    [711] }
    [712] }
    [713] }
    [714] }
    [715] }
    [716] }
    [717] }
    [718] }
    [719] }
    [720] }
    [721] }
    [722] }
    [723] }
    [724] }
    [725] }
    [726] }
    [727] }
    [728] }
    [729] }
    [730] }
    [731] }
    [732] }
    [733] }
    [734] }
    [735] }
    [736] }
    [737] }
    [738] }
    [739] }
    [740] }
    [741] }
    [742] }
    [743] }
    [744] }
    [745] }
    [746] }
    [747] }
    [748] }
    [749] }
    [750] }
    [751] }
    [752] }
    [753] }
    [754] }
    [755] }
    [756] }
    [757] }
    [758] }
    [759] }
    [760] }
    [761] }
    [762] }
    [763] }
    [764] }
    [765] }
    [766] }
    [767] }
    [768] }
    [769] }
    [770] }
    [771] }
    [772] }
    [773] }
    [774] }
    [775] }
    [776] }
    [777] }
    [778] }
    [779] }
    [780] }
    [781] }
    [782] }
    [783] }
    [784] }
    [785] }
    [786] }
    [787] }
    [788] }
    [789] }
    [790] }
    [791] }
    [792] }
    [793] }
    [794] }
    [795] }
    [796] }
    [797] }
    [798] }
    [799] }
    [800] }
    [801] }
    [802] }
    [803] }
    [804] }
    [805] }
    [806] }
    [807] }
    [808] }
    [809] }
    [810] }
    [811] }
    [812] }
    [813] }
    [814] }
    [815] }
    [816] }
    [817] }
    [818] }
    [819] }
    [820] }
    [821] }
    [822] }
    [823] }
    [824] }
    [825] }
    [826] }
    [827] }
    [828] }
    [829] }
    [830] }
    [831] }
    [832] }
    [833] }
    [834] }
    [835] }
    [836] }
    [837] }
    [838] }
    [839] }
    [840] }
    [841] }
    [842] }
    [843] }
    [844] }
    [845] }
    [846] }
    [847] }
    [848] }
    [849] }
    [850] }
    [851] }
    [852] }
    [853] }
    [854] }
    [855] }
    [856] }
    [857] }
    [858] }
    [859] }
    [860] }
    [861] }
    [862] }
    [863] }
    [864] }
    [865] }
    [866] }
    [867] }
    [868] }
    [869] }
    [870] }
    [871] }
    [872] }
    [873] }
    [874] }
    [875] }
    [876] }
    [877] }
    [878] }
    [879] }
    [880] }
    [881] }
    [882] }
    [883] }
    [884] }
    [885] }
    [886] }
    [887] }
    [888] }
    [889] }
    [890] }
    [891] }
    [892] }
    [893] }
    [894] }
    [895] }
    [896] }
    [897] }
    [898] }
    [899] }
    [900] }
    [901] }
    [902] }
    [903] }
    [904] }
    [905] }
    [906] }
    [907] }
    [908] }
    [909] }
    [910] }
    [911] }
    [912] }
    [913] }
    [914] }
    [915] }
    [916] }
    [917] }
    [918] }
    [919] }
    [920] }
    [921] }
    [922] }
    [923] }
    [924] }
    [925] }
    [926] }
    [927] }
    [928] }
    [929] }
    [930] }
    [931] }
    [932] }
    [933] }
    [934] }
    [935] }
    [936] }
    [937] }
    [938] }
    [939] }
    [940] }
    [941] }
    [942] }
    [943] }
    [944] }
    [945] }
    [946] }
    [947] }
    [948] }
    [949] }
    [950] }
    [951] }
    [952] }
    [953] }
    [954] }
    [955] }
    [956] }
    [957] }
    [958] }
    [959] }
    [960] }
    [961] }
    [962] }
    [963] }
    [964] }
    [965] }
    [966] }
    [967] }
    [968] }
    [969] }
    [970] }
    [971] }
    [972] }
    [973] }
    [974] }
    [975] }
    [976] }
    [977] }
    [978] }
    [979] }
    [980] }
    [981] }
    [982] }
    [983] }
    [984] }
    [985] }
    [986] }
    [987] }
    [988] }
    [989] }
    [990] }
    [991] }
    [992] }
    [993] }
    [994] }
    [995] }
    [996] }
    [997] }
    [998] }
    [999] }
    [1000] }

```

Fig.2

3/13

300 →

302 →

```

decl g;

main()
begin
  decl h;
  h := !g;
[6]   A(g,h);
[7]   skip;
[8]   A(g,h);
[9]   skip;
[10]  if (g) then
[11]  R: skip;
[12]  else
[14]  skip;
      fi
    end

  A(a1,a2)
  begin
[20]  if (a1) then
[21]  A(a2,a1);
[22]  skip;
      else
[24]  g := a2;
      fi
    end
  end

```

→ 304

bebop v1.0: (c) Microsoft Corporation.
 Done creating bdd variables
 Done building transition relations

Label R reachable by following path:

Line 12	State g=1 h=0
Line 11	State g=1 h=0
Line 10	State g=1 h=0
Line 22	State g=1 a1=1 a2=0
Line 24	State g=1 a1=0 a2=1
Line 20	State g=1 a1=0 a2=1
Line 21	State g=1 a1=1 a2=0
Line 20	State g=1 a1=1 a2=0
Line 9	State g=1 h=0
Line 8	State g=1 h=0
Line 22	State g=1 a1=1 a2=0
Line 24	State g=1 a1=0 a2=1
Line 20	State g=1 a1=0 a2=1
Line 21	State g=1 a1=1 a2=0
Line 20	State g=1 a1=1 a2=0
Line 7	State g=1 h=0
Line 6	State g=1

Fig.3

FOUO 06099860

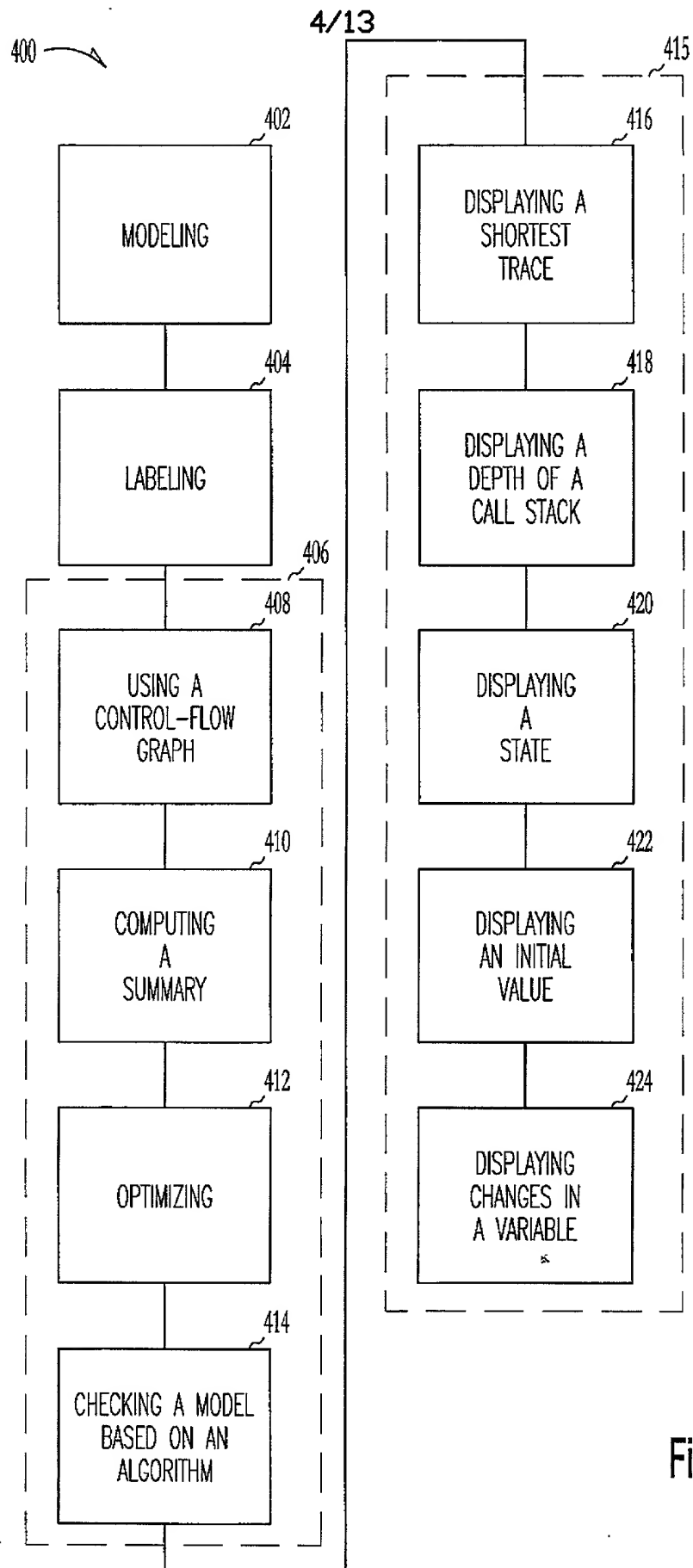


Fig.4

5/13

500

```
[1]  if (z) {  
[2]    x:=1;  
      else  
[3]    x:=z;  
[4]    z:=y|x;
```

Fig.5A

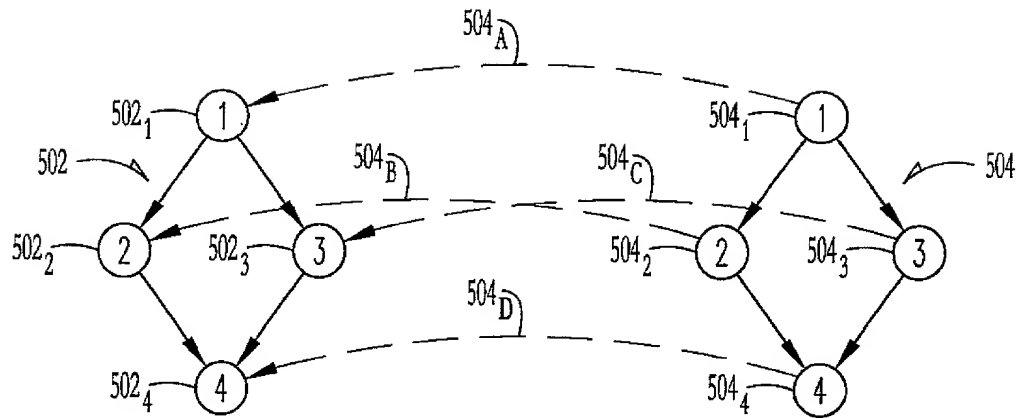


Fig.5B

6/13

600

```
Boolean g;  
main() {  
  [2]   if (z) {  
  [3]     x:=1;  
  [4]     else  
          x:=0;  
  [5]     z:=y+x;  
  [6]     foo (z);  
  [7]     skip;  
  [8]   }  
  [9]   foo (z) {  
  [10]    g:=1;  
  [11]  }  
}
```

Fig.6A

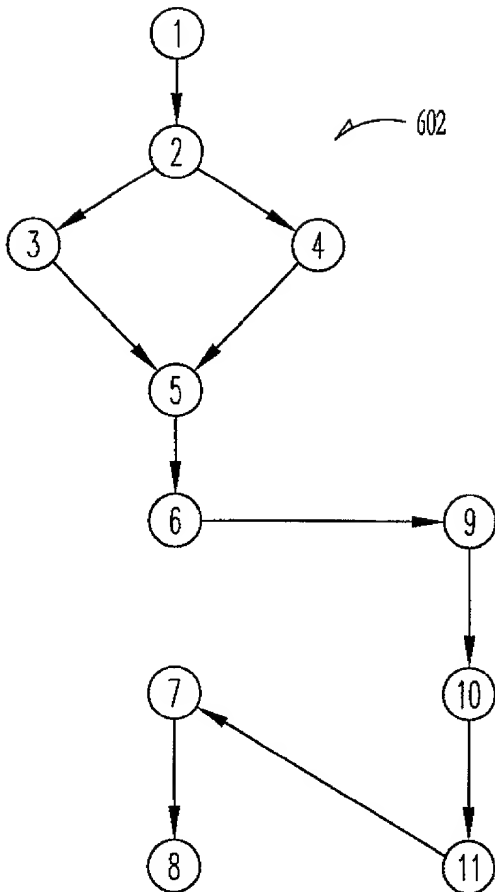


Fig.6B

7/13

700

v	$Transfer_v$
702 skip print goto return	$\lambda(\Omega_1, \Omega_2).(\Omega_2 = \Omega_1)$
704 $x_1, \dots, x_k :=$ e_1, \dots, e_k	$\lambda(\Omega_1, \Omega_2).(\Omega_2 = \Omega_1 [x_1/\Omega_1(e_1)] \dots [x_k/\Omega_1(e_k)])$
706 if(d) while(d) assert(d)	$Transfer_{v,true} = \lambda(\Omega_1, \Omega_2).((\Omega_1(d) = 1) \wedge (\Omega_2 = \Omega_1))$ $Transfer_{v,false} = \lambda(\Omega_1, \Omega_2).((\Omega_1(d) = 0) \wedge (\Omega_2 = \Omega_1))$
708 pr(e_1, \dots, e_k)	$\lambda(\Omega_1, \Omega_2).(\Omega_2 = \Omega_1 [x_1/\Omega_1(e_1)] \dots [x_k/\Omega_1(e_k)]),$ where x_1, \dots, x_k are the formal parameters of pr

Fig.7

0986090.05301

8/13

800

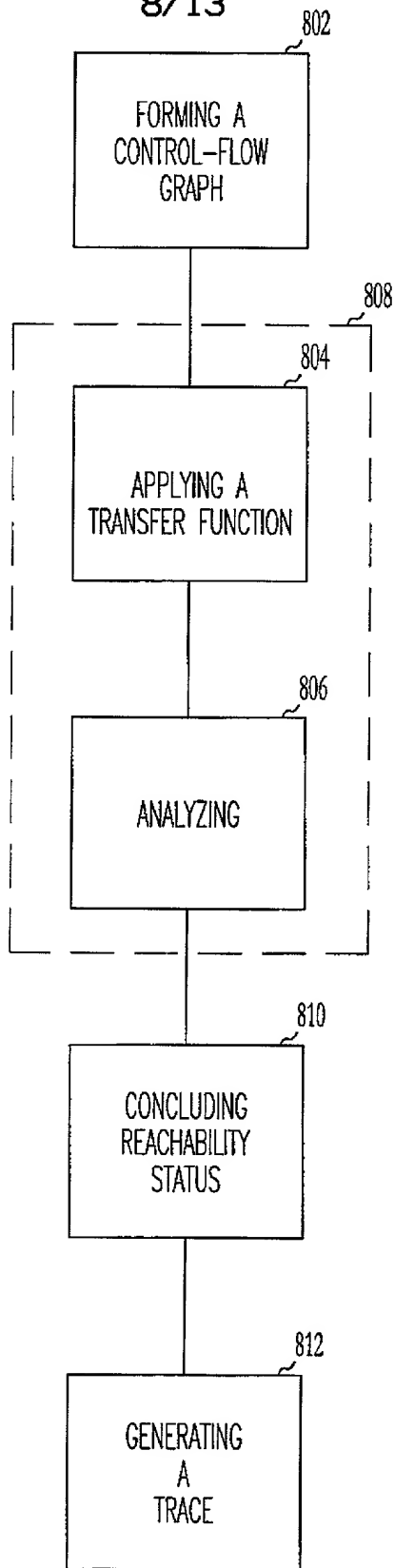


Fig.8

9/13

```

902 global PathEdges, SummaryEdges, WorkList 900
904 procedure Propagate(v, p)
    begin
906 if  $p \not\subseteq \text{PathEdges}(v)$  then
908  $\text{PathEdges}(v) := \text{PathEdges}(v) \cup p$ 
910 Insert  $v$  into WorkList fi
    fi
912 end

914 procedure Reachable( $G_B$ )
    begin
916 for all  $v \in V_B$  do  $\text{PathEdges}(v) := \{\}$ 
917 for all  $v \in \text{Call}_B$  do  $\text{SummaryEdges}(v) := \{\}$ 
918  $\text{PathEdges}(\text{First}_B(\text{main})) :=$ 
         $\{\langle \Omega, \Omega \rangle \mid \Omega \text{ is any valuation to globals and local variables of main} \}$ 
920  $\text{WorkList} := \{\text{First}_B(\text{main})\}$ 
922 while  $\text{WorkList} \neq 0$  do
924 remove vertex  $v$  from WorkList
926 switch ( $v$ )
928 case  $v \in \text{Call}_B$ 
         $\text{Propagate}(\text{sSucc}_B(v), \text{SelfLoop}(\text{Join}(\text{PathEdges}(v), \text{Transfer}_v)))$  930
         $\text{Propagate}(\text{ReturnPt}_B(v), \text{Join}(\text{PathEdges}(v), \text{SummaryEdges}(v)))$  932
934 case  $v \in \text{Exit}_B$ :
        for each  $w \in \text{Succ}_B(v)$  do 936
            let
                 $c \in \text{Call}_B$  such that  $w = \text{ReturnPt}_B(c)$  and 938
                 $s = \text{Lift}_c(\text{PathEdges}(v), \text{ProcOf}_B(v))$  940
            in
                if  $s \not\subseteq \text{SummaryEdges}(c)$  then 944
                     $\text{SummaryEdges}(c) := \text{SummaryEdges}(c) \cup s$  946
                     $\text{Propagate}(w, \text{Join}(\text{PathEdges}(c), \text{SummaryEdges}(c)))$ ; 948
                ni
950 case  $v \in \text{Cond}_B$ :
                 $\text{Propagate}(\text{TSucc}_B(v), \text{Join}(\text{PathEdges}(v), \text{Transfer}_{v, \text{true}}))$  952
                 $\text{Propagate}(\text{FSucc}_B(v), \text{Join}(\text{PathEdges}(v), \text{Transfer}_{v, \text{false}}))$  954
956 case  $v \in V_B - \text{Call}_B - \text{Exit}_B - \text{Cond}_B$ :
                let  $p = \text{Join}(\text{PathEdges}(v), \text{Transfer}_v)$  in 958
                    for each  $w \in \text{Succ}_B(v)$  do 960
                         $\text{Propagate}(w, p)$  962
                    ni
    end

```

Fig.9

FOI 50 06099860

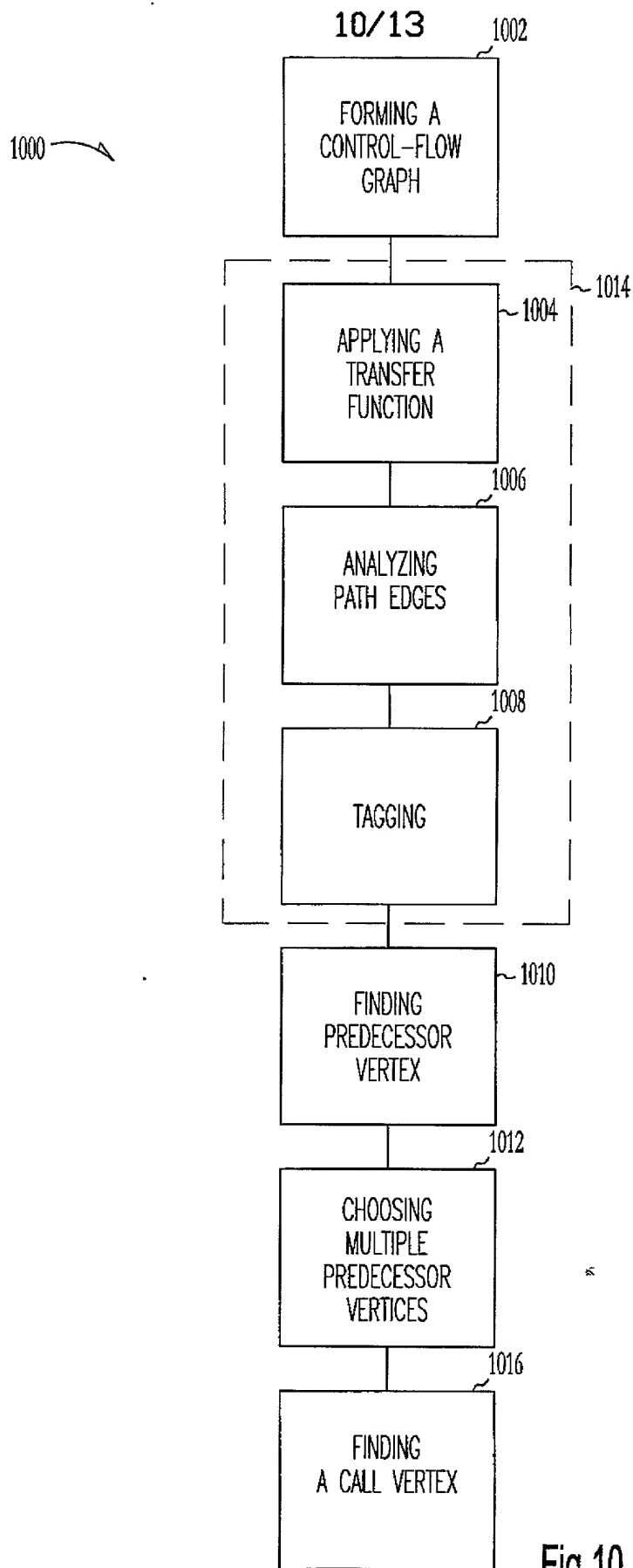


Fig.10

11/13

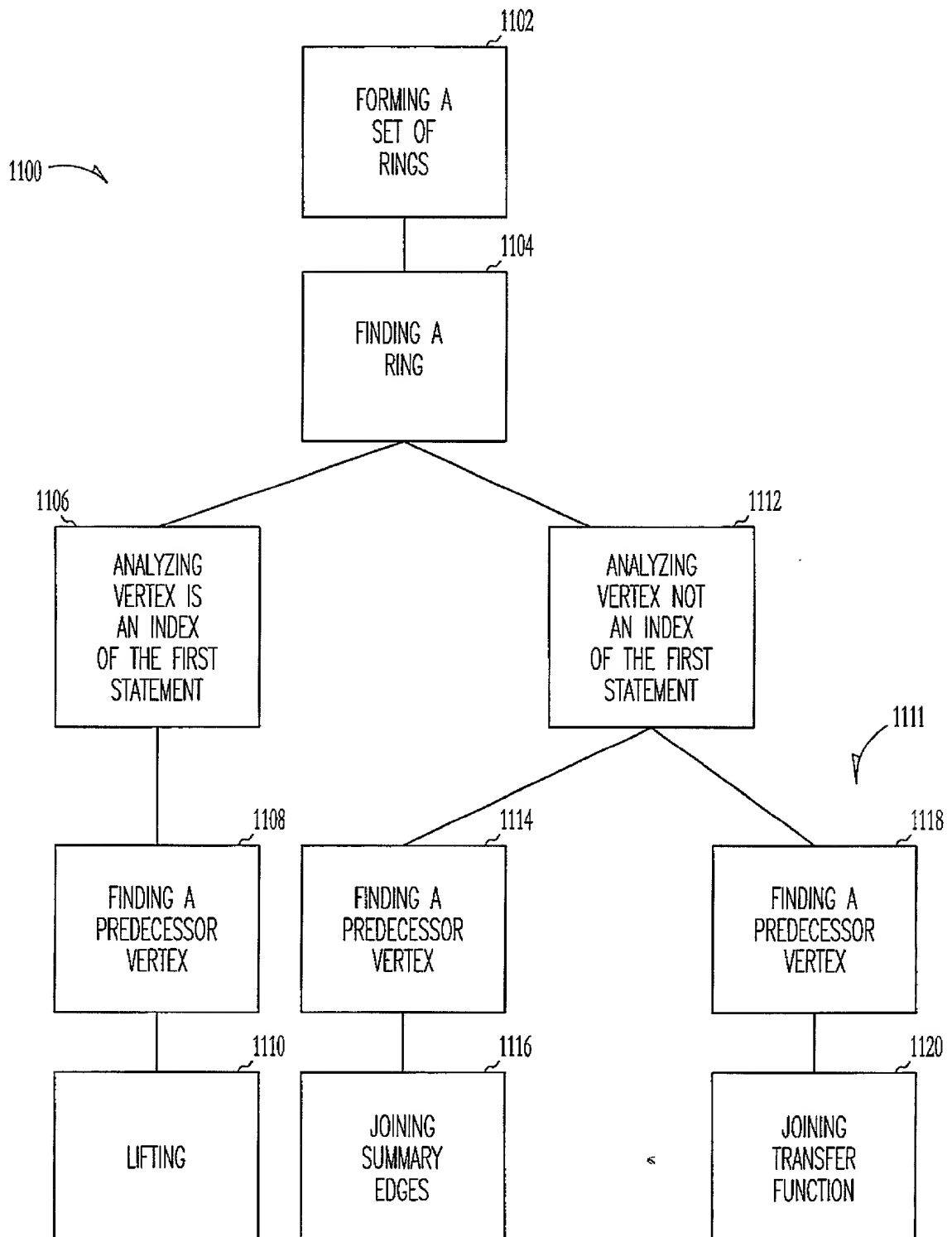


Fig.11

12/13

global

$PE' : V_G \rightarrow \text{set-of } (D \times D)$

$Worklist : V_G \rightarrow \text{set-of } (D \times D)$

procedure Propagate($v : V_{G,p} : (D \times D)$)

begin

if $p \notin PE'(v)$ **then**

$PE'(v) := PE'(v) \cup \{p\}$

$Worklist(v) := Worklist(v) \cup \{p\}$

fi

end

procedure CMOP_{SP_{rhs}}($S : \text{set-of } D$)

begin

$PE'(\text{entry}) := \{\langle d, d \rangle \mid d \in S\}$

$Worklist(\text{entry}) := PE'(\text{entry})$

while $\exists v_2 \text{ s.t. } Worklist(v_2) \neq \emptyset$ **do**

 select and remove $\langle d_1, d_2 \rangle$ from $Worklist(v_2)$

for each $v_2 \rightarrow v_3 \in E_G$ **do**

for each $d_3 \in M(v_2 \rightarrow v_3)(\{d_2\})$ **do**

 Propagate($v_3 \langle d_1, d_3 \rangle$)

od

od

od

end

Fig.12

13/13

global
 $PE' : V \rightarrow \text{set-of}(\text{set-of } D \times \text{set-of } D)$
 $Worklist : V_G \rightarrow \text{set-of}(\text{set-of } D \times \text{set-of } D)$

procedure Propagate($v ; V_G, p : (\text{set-of } D \times \text{set-of } D)$)
begin
 if $p \notin PE'(v)$ then
 $PE'(v) := PE'(v) \cup \{p\}$
 $Worklist(v) := Worklist(v) \cup \{p\}$
 fi
end

procedure CSMOP_{SP_{rhs}}($S' : \text{set-of}(\text{set-of } D)$)
begin
 $PE'(\text{entry}) := \{\langle S, S \rangle \mid S \in S'\}$
 $Worklist(\text{entry}) := PE'(\text{entry})$
while $\exists v_2 \text{ s.t. } Worklist(v_2) \neq \emptyset$ **do**
 select and remove $\langle S_1, S_2 \rangle$ from $Worklist(v_2)$
for each $v_2 \rightarrow v_3 \in E_G$ **do**
 let $S_3 = M(v_2 \rightarrow v_3)(S_2)$ **in**
 Propagate($v_3, \langle S_1, S_3 \rangle$)
ni
od
od
end

Fig.13